|  |  |
| --- | --- |
| **Categorías Léxicas** | |
| **Reservadas** | If, else, else if, while, for, int, string, double |
| **Operadores** | +, -, \*, /, >, <, &, !, | |
| **Asignación** | = |
| **Comentarios** | //, /\*, \*/ |
| **Entero** | 0-9 |
| **Identificador** | Palabra2, i |
| **Blanco** | Espacio, línea nueva, tab |
| **Terminador** | ; |
| **Símbolos Especiales** | (, ), {, } |
| **EOF** | Eof |
| **Constantes de carácter** | “cadenas” |
| **Real** | 0.7 |

TOKENS:

[0] = [error]

[1] = [if]

[2] = [else]

[3] = [else if]

[4] = [while]

[5] = [for]

[6] = [int]

[7] = [string]

[8] = [double]

[9] = [+]

[10] = [-]

[11] = [\*]

[12] =[/]]

[13] =[>]

[14] = [<]

[15] = [&]

[16] = [!]

[17] = [|]

[18] = [=]

[19] = [<=]

[20] = [>=]

[21] = [#]

[22] = [$@]

[23] = [@$]

[24] = [;]

[25] = [(]

[26] = [)]

[27] = [{]

[28] = [}]

[29] = [ ]

[30] = [/s]

[31] = [/n]

[32] = [/t]

Analizador sintactico

Sent -> cond

Sent -> ciclo

Sent -> Comentario

Sent -> Tipo

Sent -> EoF

Tipo -> int ID = [0-9]+ Terminador

Tipo -> string ID = “[.\*]“ Terminador

Tipo -> double ID = [0-9]\* (\.[0.9]\*)? Terminador

Operador -> +

Operador -> -

Operador -> \*

Operador -> /

Comparador -> >

Comparador -> <

Comparador -> &

Comparador -> !

Comparador -> |

Comparador -> =

Comparador -> <=

Comparador -> >=

Cond -> if(expresion comparador expresion){ sent } SubCond

SubCond ->

SubCond -> else { sent }

SubCond -> elseif(expresion comparador expresion { sent } SubCond

Ciclo -> while(expresion comparador expresion){ sent }

Ciclo -> for(int asig; ID comparador expr; ID++){ sent }

Asig -> ID = [0-9]+

Asig -> ID = “.\*”

Asig -> ID = [0-9]\* (\.[0.9]\*)?

Comentario -> # sent Salto

Comentario -> ComentarioInicio sent ComentarioFinal

ComentarioInicio -> $@

ComentarioFinal -> @$

Terminador -> ;

Espacio -> [/s]

Salto -> [/n]

Tabulacion -> [/t]

EoF -> eof

ID -> [a-z A-Z] [a-z A-Z 0-9]\*